IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Ian J. Deverill, et al.	
Application No.: 09/896,854	Group Art Unit: 2154
Filed: June 29, 2001	Examiner: H. Patel
For: SYSTEM AND METHOD TO MEASURE LATENCY OF TRANSACTION INFORMATION FLOWING THROUGH A COMPUTER SYSTEM	
Attorney Docket No.: 11252-0009	

I, John F. Letchford, Registration No. 33,328, certify that this correspondence is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to Mail Stop Appeal Brief - Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on February 16, 2005.

John F. Letchford

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

APPELLANTS' BRIEF PURSUANT TO 37 CFR § 1.192

The above-identified application comes before the United States Patent and Trademark Office ("USPTO") Board of Appeals and Interferences ("Board") from a Final Rejection of claims 1-12 dated November 8, 2004.

I. REAL PARTY IN INTEREST

The real party in interest in the present appeal is Goldman, Sachs & Company, a partnership formed under the laws of the State of New York with its principal place of business located at 85 Broad Street, New York, NY 10004, USA (hereinafter "Assignee"), as evidenced by an assignment of the entire right, title and interest in and to the application from the inventors, Ian J. Deverill, Safe E. Hammad, Alex D. Hassan and Warren P. Finnerty, to Assignee, which is recorded in the USPTO at reel 012109 and frame 0231.

II. RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences known to Appellants, Assignee or the undersigned which will directly affect, be directly affected by or have a bearing on the Board's decision in the presently pending appeal.

III. STATUS OF THE CLAIMS

The status of the claims in the application is as follows:

Claims 1-12 remain in the application and are finally rejected.

IV. STATUS OF AMENDMENTS FILED SUBSEQUENT TO THE FINAL REJECTION

No amendments were filed subsequent to the Final Rejection.

V. SUMMARY OF THE INVENTION

Most broadly, the invention defined in the claims on appeal is addressed to methods, application program interfaces and systems for monitoring the performance of computer systems using transaction latency data. The claims on appeal include three (3) independent claims, claims 1, 6 and 9.

The method of monitoring a computer application recited in independent claim 1 on appeal involves the steps of:

without predefining events describing the potential stages of a transaction to be executed by said computer application, adding software code to said computer application for assigning a single general reference to characteristic transactional information associated with said transaction;

using said single general reference to identify transaction events performed by said computer application in executing said transaction; and

measuring said transactions.

Claims 2-5 on appeal further enlarge upon the method of monitoring a computer application executed on a computer system defined in independent claim 1 to define various features that are believed to be representative of preferred aspects thereof.

The application program interface for use in monitoring a computer application executed on a computer system recited in independent claim 6 on appeal involves:

software code added to said computer application for assigning, without predefining events describing the potential stages of a transaction to be executed by said computer application, a single general reference to characteristic transactional information associated with a transaction to be executed by said computer application; and

an agent for marking the time at which said software code is executed and tagging that time with said characteristic transactional information as said characteristic transactional information is being currently processed by the computer application.

Claims 7 and 8 on appeal further enlarge upon the application program interface for use in monitoring a computer application executed on a computer system defined in independent claim 6 to define various features that are believed to be representative of preferred aspects thereof.

The computer system performance monitoring system recited in independent claim 9 on appeal involves (with appropriate drawing figure reference numerals provided wherever possible to facilitate understanding of the invention):

an application program interface for monitoring a computer application executed on a computer system (600), said application program interface comprising:

software code added to said computer application for assigning, without predefining events describing the potential stages of a transaction to be executed by said

computer application, a single general reference to characteristic transactional information associated with a transaction to be executed by said computer application; and

an agent for marking the time at which said software code is executed and tagging that time with said characteristic transactional information as said characteristic transactional information is being currently processed by the computer application;

an aggregator (612) for calculating computer application latency data from raw timing data produced by said agent; and

a database (610) for storing said raw computer application timing data and said latency data.

Claims 10-12 on appeal further enlarge upon the computer system performance monitoring system defined in independent claim 9 to define various features that are believed to be representative of preferred aspects thereof.

The present invention employs unique methods and apparatus, including a novel application program interface (API), for measuring the precise latency of information flowing through computer systems regardless of system topology. The computer systems whose information latency may be monitored by the invention range from individual computers through wide area networks comprising thousands of computers. The metrics measured by the present invention include the processing time for a given transaction event within each computing process, the time spent

by the transaction between each computing process, and the time taken by the transaction to pass through the entire computer system. A "transaction event" may include a request, a response, a directive not requiring a response, a complete or partially complete subtransaction, an automatically or manually triggered computer function, a database store or retrieve function or any other time-monitorable function or calculation that may be performed by a computer system.

The present invention involves the association of measurement or timing data with inherent transactional information that is ordinarily passed from one component of a computer system to the next in the course of executing a transaction by a computer application. The raw measurement or timing data is then used to calculate latency data for some or all aspects of a particular transaction. Significantly, the invention performs these operations based solely on references drawn only from the business or other transactional data associated with the transaction.

According to the invention, computer systems may be instrumented using an uncomplicated and minimally invasive API that allows the execution time of every task in an application to be measured and transactional information to be tagged to the measurement or timing data. Preferably, the measurement data is also associated with related measurement data from other components or systems of the same business entity in order to give precise latency data and statistics for transactions that involve more then one segment of a business' operations. By way of example, a trade reference or other inherent conventional business information normally associated with an electronic

business transaction may be used as the transaction identifier throughout the latency measurement process. Then, individual tasks required for processing the task can be tagged with timing data whereby the data can be processed and evaluated to produce desired latency and/or summary data. For instance, the average, maximum or minimum time taken to process commodities or securities trades from front-office to back-office within a brokerage or investment house may be calculated with precision in order to obtain meaningful information concerning the efficiency of the business entity's computerized transaction system.

With the present invention, no special languages, software code and rules must be created to predefine and pre-classify events and transactions. Furthermore, no transaction handle or correlator needs to be created for a transaction and passed from one component of a computer system to the next since the business information normally associated with the transaction is itself sufficient to identify and track the transaction. Since no use is made of API-generated handles, the present system easier to deploy in modular systems than conventional application response measurement (ARM) systems.

Moreover, multiple computer systems or components that may be required to execute an entire transaction need not be confined to being client-server in nature since the API according to the invention can be mapped to other topologies, e.g., a distributed system. The technique is simple in design and implementation, minimally invasive, and highly scalable in order to accommodate potentially large volumes and frequencies of information flow through vast computer systems and networks.

Among its advantages, the present system eliminates guess-work from computer system capacity estimates, enables ready assessment of the performance impact of new computer application releases and migrations, identifies application performance trends, works with either intra-system or inter-system IT system optimization tools, assists in researching user performance complaints and generates system management reports.

Presently existing computer system latency monitoring system assume a wide variety of configurations and/or methodologies. Several of these designs were discussed at length in the specification of the present application as well as during prosecution thereof. Regardless of their individual constructions, indeed, because of their structural/operational peculiarities, none of these systems, including those disclosed in the references cited against the claims on appeal, is capable of producing the advantages afforded by the claimed invention now before the Board.

As will be clearly demonstrated hereinafter, the invention defined in all of the claims on appeal is neither disclosed nor suggested, either expressly or implicitly, by the references relied upon by the Examiner, whether those references are considered individually or in any conceivably combination.

VI. ISSUES

A statement of each separate ground of objection or rejection Appellant wishes to be reviewed, including the basis of each ground of rejection is as follows:

(1) The title of the invention is objected to for allegedly not being descriptive, i.e., is not clearly indicative of the invention to which the claims are directed.

(2) Claims 1, 2, 6 and 7 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Leymann et al. (U.S. Patent No. 6,633,908, "Leymann").

(3) Claims 3, 8 and 9 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Leymann in view of Maccabee et al. (U.S. Patent No. 6,108,700, "Maccabee"), whereas claims 4, 10 and 11 stand rejected over Leymann and Maccabee under 35 U.S.C. § 103(a) in view of "Official Notice."

(4) Claims 5 and 12 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Leymann and Maccabee in view of "Official Notice."

VII. GROUPING OF CLAIMS

Appellant wishes to group claims 1, 2, 6 and 7; claims 3, 4 and 8-11; and claims 5 and 12 as three (3) distinct groups of claims to stand or fall separately by this appeal.

VIII. ARGUMENT

(1) Objection to the Title of the Invention

The amended title of the invention submitted with the Amendment filed August 9, 2004 is objected to for allegedly not being descriptive, i.e., is not clearly indicative of the

invention to which the claims are directed. Such objection is respectfully traversed.

The amended title is: SYSTEM AND METHOD TO MEASURE LATENCY OF TRANSACTION INFORMATION FLOWING THROUGH A COMPUTER SYSTEM. At Section 3, "Specification" at page 5 of the final Office Action, the Examiner suggested the following title: MEASURING LATENCY OF TRANSACTION FLOWING THROUGH AN END-TO-END NETWORK COMPUTER SYSTEMS REGARDLESS OF NETWORK TOPOLOGY. The Examiner suggested a similar title in the first Office Action dated February 12, 2004. Appellants respectfully decline to adopt either title suggested by the Examiner.

The lengthy titles suggested by the Examiner are believed to be overly detailed and directed to limited instantiations of the present invention. For example, the first paragraph of the "SUMMARY OF THE INVENTION" section at page 6 of Appellants' specification makes it clear that the "computer systems" that may be monitored by the present invention may range from a single computer to an end-to-end network computer system. The Examiner's suggested titles relate only to "end-to-end network computer systems regardless of network topology" (emphasis added). As a compromise, Appellants offered the amended title of the invention set forth hereinabove. Such title accurately captures all aspects and instantiations of the present invention. Accordingly, acceptance of that title is respectfully requested.

(2) Rejection of Claims 1, 2, 6 and 7 under 35 U.S.C. § 102(e)

Claims 1, 2, 6 and 7 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Leymann et al. (U.S. Patent No. 6,633,908, "Leymann"). Such rejection is respectfully traversed.

For the Board's convenience, independent claims 1 and 6 on appeal are reproduced herebelow, with emphasis added.

> 1. A method of monitoring a computer application executed on a computer system, said method comprising the steps of:
>
> without predefining events describing the potential stages of a transaction to be executed by said computer application, adding software code to said computer application for assigning a single general reference to characteristic transactional information associated with said transaction;
>
> using said single general reference to identify transaction events performed by said computer application in executing said transaction; and
>
> measuring said transactions.

> 6. An application program interface for use in monitoring a computer application executed on a computer system comprised of at least one component, said application program interface comprising:
>
> software code added to said computer application for assigning, without predefining events describing the potential stages of a transaction to be executed by said computer application, a single general reference to characteristic transactional information associated with a transaction to be executed by said computer application; and

> an agent for marking the time at which said software code is executed and tagging that time with said characteristic transactional information as said characteristic transactional information is being currently processed by the computer application.

In the third indented subparagraph under paragraph 6 at page 6 of the final Office Action, the Examiner states that Leymann teaches "adding software code to said computer application (e.g., inherent, for example, addition of software code to the computer application when the computer application is created ...)." With due deference, with regard to this particular feature of Appellants' claimed invention, what the Examiner contends is "inherent" in Leymann is most certainly not. Indeed, the particular passage in Leymann relied upon by the Examiner to substantiate his position (as well as several others) specifically instruct one <u>not</u> to add software to the computer application under scrutiny.

> The present invention enables the management and measurement of application performance in systems management environments <u>without special instrumentation of the corresponding applications</u>.
>
> <u>The basic idea of the present invention is not to instrument the application components. The present invention contemplates instrumenting the invocation agent instead, which in turn is responsible for calling the application for execution. This solution provides application response measurement without any modification of the application being measured. As a consequence, no special code has to be added to newly or existing applications to enable them for response measurement</u>.

Leymann at col. 3, lines 5-17 (emphasis added).

> <u>The basic idea of the present invention is to instrument not the application components. The present</u>

> invention contemplates instrumenting the invocation agent instead, which in turn is responsible for calling the application for execution.

Leymann at col. 7, line 66, through column 8, line 1 (emphasis added).

> The present invention relates to the area of systems management teaching means and a method for determining and managing application performance. Application Response Measurement (ARM) assumes that the managed application is a self-instrumented component. This requires invasive changes to existing applications or to add explicitly code to newly written applications. Due to this additional effort this will restrict the area of applicability of ARM. The basic idea of the present invention is to instrument not the application components. The present invention contemplates instrumenting the invocation agent instead, which in turn is responsible to call the application for execution. This solution provides application response measurement without any modification of the application being measured. It is the invocation agent that makes the appropriate ARM calls to furnish the instrumentation on behalf of the application.

Leymann at Abstract (emphasis added).

Each of independent claims 1 and 6 (as well as independent claim 9, *infra*) expressly require that software code be added to the computer application being monitored by the API of the present invention to assign a single general reference to characteristic transactional information associated with a computer application transaction under surveillance. In stark contrast, Leymann does so by way of a separate application invocation agent. Leymann is thus diametrically opposed in architecture and operation to the system and method recited in Appellants' independent claims 1 and 6.

Lastly, in the "Response to Arguments" section at page 2, paragraph 2, of the final Office Action, the Examiner states (with underscoring in original text, italics added):

> "[I]t is noted that the features upon which applicant relies 'software code is added to the computer application being monitored by the API to assign a single general reference to characteristic transactional information associated with a computer application transaction under surveillance' *are not recited in the rejected claim(s)*. Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims."

As plainly shown hereinabove and in the underscored additions to independent claims 1, 6 and 9 submitted with the Amendment filed August 9, 2004, the notion of adding software code to a computer application being monitored in order to assign a single general reference to characteristic transactional information associated with the application is expressly recited in claims 1, 6 and 9. The Examiner's assertion that is not is bewildering. Moreover, it is in irreconcilable conflict with the his failed efforts to divine such a teaching from the Leymann patent. Query: if this limitation were not present in the claims presented by the August 9, 2004 Amendment, then why did the Examiner strive as he did to find it in the prior art?

For the foregoing reasons, Appellants submit that Leymann fails to anticipate the present invention as claimed in independent claims 1 and 6 and, derivatively, their respective dependent claims 2 and 7. Indeed, in critical respects, Leymann leads one of ordinary skill in the art directly away therefrom. Accordingly, Appellants kindly submit that the outstanding

rejection of claims 1, 2, 6 and 7 under 35 U.S.C. 102(e) as being anticipated by Leymann is improper and should be reversed.

(3) Rejection of Claims 3, 4 and 8-11 under 35 U.S.C. § 103(a)

Claims 3, 8 and 9 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Leymann in view of Maccabee (U.S. Patent No. 6,108,700 "Maccabee"). Claims 4, 10 and 11 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Leymann and Maccabee in view of "Official Notice." Such rejections are respectfully traversed.

As demonstrated above, no combination of the teachings of the Leymann patent with those of the Maccabee patent will produce either of the inventions prescribed in Appellants' independent claims 1 and 6. Like those claims, independent claim 9 also includes a unique combination of features that are not taught or suggested by the Leymann and Maccabee patents, whether considered alone or in combination.

Accordingly, Appellants' remarks concerning the inapplicability of the combination of the Leymann-Maccabee reference tandem to independent claims 1 and 6 is hereby adopted, incorporated and reasserted in its entirety in connection with independent claim 9.

Further, notwithstanding what Maccabee may or may not disclose in relation to Appellants' claims 3, 4, 8, 10 and 11, Maccabee teaches directly away from independent claims 1, 6 and 9. The disadvantages of the Maccabee system were addressed at

length at page 5 of Appellants' specification as originally filed. In short, Maccabee proposes the creation of a transaction definition language called the ETE (End-to-End) Transaction Definition Language that specifies how to construct identifiable transactions from events and links. In an illustrated example, the ETE Transaction Definition Language disclosed in Maccabee requires the creation of twenty-one (21) lines of software code merely to define something as relatively simple as a Web commerce transaction. Merely contemplating all of the possible events and transactions that might be involved in a complex business transaction, particularly one whose execution involves the coordination of several business entities and computer systems, is itself a daunting task. Codifying these items complicates the task. That is, individually defining -- in advance -- all of these possible events and transactions in software code in order to produce a complete set of transaction generation rules amounts to a potentially vast amount of preliminary preparation activity that must be performed before the monitoring system may be placed into operation.

The present invention requires no predefinition of events and thereby avoids the disadvantages of the Maccabee system. This beneficial feature is present in each of independent claims 1, 6 and 9. The pertinent passage from claim 9 is representative:

> software code added to said computer application for assigning, without predefining events describing the potential stages of a transaction to be executed by said computer application, a single general reference to characteristic transactional information associated with a transaction to be executed by said computer application;

Maccabee is therefore in direct conflict with an essential feature of each Appellants' independent claims 1, 6 and 9. Accordingly, since Maccabee leads one directly away from the invention prescribed in those claims, it necessary also leads one directly away from the invention defined by dependent claims 3, 4, 8 10 and 11, which variously depend from independent claims 1, 6 and 9. As a result, since no combination of the teachings of Leymann and Maccabee can produce Appellants' invention as defined by independent claims 1, 6 and 9, no combination of the Leymann-Maccabee reference tandem, whether considered alone or in combination with "Official Notice" can render obvious dependent claims 3, 4, 8, 10 and 11.

(4) Rejection of Claims 5 and 12 under 35 U.S.C. § 103(a)

Claims 5 and 12 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Leymann and Maccabee in view of "Official Notice." Such rejection is respectfully traversed.

Whether considered individually or collectively, Leymann and Maccabee do not suggest the novel inventions of independent claims 1 and 9 from which claims 5 and 12 respectively depend. Therefore, they do not disclose or suggest the unique method of monitoring a computer system or a monitoring system for doing same as particularly defined in Appellants' claims 5 and 12. Furthermore, the Examiner has failed to show -- other than unsubstantiated "Official Notice" -- any tangible teaching or suggestion of calculating latency of transactional information passed between components of computer systems according to the formula:

$$\frac{(T'_1(U_{Cy}) - T'_1(V_{Cx})) + (T'_2(U_{Cy}) - T'_2(V_{Cx})) + \ldots + (T'_{m-1}(U_{Cy}) - T'_{m-1}(V_{Cx})) + (T'_m(U_{Cy}) - T'_m(V_{Cx}))}{m}$$

where:

m = an unspecified number of transaction events, T_1, T_2, ..., T_{m-1}, T_m;

T'_1, T'_2, ..., T'_{m-1}, T'_m = transactional information pertaining to transaction events, T_1, T_2, ..., T_{m-1}, T_m;

U_{Cy} = start time for a transaction event at one component of said computer system; and

V_{Cx} = end time for a transaction event at another component of said computer system.

Therefore, it is not lightly to be assumed that the asserted "Official Notice" provides sufficient basis upon which the Examiner may rely for specific or implied suggestion that the foregoing formula for calculating latency of transactional information passed between components of a computer system is so well known that "Official Notice" renders the formula generally known or obvious to those skilled in the art of computer application latency measurement.

Accordingly, Appellants kindly submit that the outstanding rejection of claims 5 and 12 under 35 U.S.C. 103(a) as being unpatentable over Leymann in view of Maccabee in view of "Official Notice" is improper and should be reversed.

To conclude, Appellants' claims must be interpreted fairly and accurately. Additionally, the teachings of the prior art

cited against the claims on appeal must be fairly and accurately interpreted for what they in fact disclose and/or suggest. The disclosures of the cited references, when so interpreted, do not disclose or suggest Appellants' claimed invention. Therefore, the invention as a whole would not have been considered anticipated or obvious to one skilled in this art at the time of Appellants' invention in light of the art relied upon by the Examiner. Accordingly, it is respectfully submitted that the Final Rejection of claims 1-12 should be reversed.

Respectfully submitted,

Date: February 16, 2005

John F. Letchford
Registration No. 33,328

KLEHR, HARRISON, HARVEY,
BRANZBURG & ELLERS LLP
260 South Broad Street
Philadelphia, PA 19102-5003
Tel.: (215) 569-3495
Fax: (215) 568-6603
e-mail: jletchford@klehr.com

IX. APPENDIX

The claims on appeal are as follows:

1. A method of monitoring a computer application executed on a computer system, said method comprising the steps of:

without predefining events describing the potential stages of a transaction to be executed by said computer application, adding software code to said computer application for assigning a single general reference to characteristic transactional information associated with said transaction;

using said single general reference to identify transaction events performed by said computer application in executing said transaction; and

measuring said transactions.

2. The method of claim 1 further comprising assigning a component-specific reference to said single general reference at each component of said computer system, said component-specific reference representing said characteristic transactional information as said computer application is executed on said computer system.

3. The method of claim 1 wherein said step of measuring comprises measuring the processing time spent by said computer application at each component of said computer system and measuring the processing time spent by said computer application between each component of said computer system.

4. The method of claim 3 further comprising the step of charting latency of said computer system over a selected time frame.

5. The method of claim 1 further comprising calculating latency of transactional information passed between components of said computer system according to the formula:

$$\frac{(T'_1(U_{Cy}) - T'_1(V_{Cx})) + (T'_2(U_{Cy}) - T'_2(V_{Cx})) + \ldots + (T'_{m-1}(U_{Cy}) - T'_{m-1}(V_{Cx})) + (T'_m(U_{Cy}) - T'_m(V_{Cx}))}{m}$$

where:

m = an unspecified number of transaction events, T_1, T_2, ..., T_{m-1}, T_m;

T'_1, T'_2, ..., T'_{m-1}, T'_m = transactional information pertaining to transaction events, T_1, T_2, ..., T_{m-1}, T_m;

U_{Cy} = start time for a transaction event at one component of said computer system; and

V_{Cx} = end time for a transaction event at another component of said computer system.

6. An application program interface for use in monitoring a computer application executed on a computer system comprised of at least one component, said application program interface comprising:

software code added to said computer application for assigning, without predefining events describing the potential stages of a transaction to be executed by said computer

application, a single general reference to characteristic transactional information associated with a transaction to be executed by said computer application; and

an agent for marking the time at which said software code is executed and tagging that time with said characteristic transactional information as said characteristic transactional information is being currently processed by the computer application.

7. The application program interface of claim 6 wherein said software code is further operable to assign a component-specific reference to said single general reference at each component of said computer system, said component-specific reference representing said characteristic transactional information as said computer application is executed on said computer system.

8. The application program interface of claim 6 wherein said agent measures the processing time spent by said computer application at each component of said computer system and measures the processing time spent by said computer application between each component of said computer system.

9. A computer system performance monitoring system comprising:

an application program interface for monitoring a computer application executed on a computer system, said application program interface comprising:

software code added to said computer application for assigning, without predefining events describing the

potential stages of a transaction to be executed by said computer application, a single general reference to characteristic transactional information associated with a transaction to be executed by said computer application; and

an agent for marking the time at which said software code is executed and tagging that time with said characteristic transactional information as said characteristic transactional information is being currently processed by the computer application;

an aggregator for calculating computer application latency data from raw timing data produced by said agent; and

a database for storing said raw computer application timing data and said latency data.

10. The computer system performance monitoring system of claim 9 further comprising a graphical user interface for enabling a user to retrieve said latency data from said database.

11. The computer system performance monitoring system of claim 10 wherein said graphical user interface includes means for charting latency of a computer system over a selected time frame.

12. The computer system performance monitoring system of claim 9 wherein said aggregator calculates latency of transactional information passed between components of said computer system according to the formula:

$$\frac{(T'_1(U_{Cy}) - T'_1(V_{Cx})) + (T'_2(U_{Cy}) - T'_2(V_{Cx})) + \ldots + (T'_{m-1}(U_{Cy}) - T'_{m-1}(V_{Cx})) + (T'_m(U_{Cy}) - T'_m(V_{Cx}))}{m}$$

where:

m = an unspecified number of transaction events, T_1, T_2, ..., T_{m-1}, T_m;

T'_1, T'_2, ..., T'_{m-1}, T'_m = transactional information pertaining to transaction events, T_1, T_2, ..., T_{m-1}, T_m;

U_{Cy} = start time for a transaction event at one component of said computer system; and

V_{Cx} = end time for a transaction event at another component of said computer system.